Adaptable Support for Programming Models in Many-core Architectures

Morten S. Rasmussen, Sven Karlsson, and Jens Sparsø

Although many-core architectures have emerged, it is unclear how these should be organized and programmed. We propose a tile-based many-core architecture where programmable network interfaces ties processors, local memories, caches and the on-chip interconnect together and provide basic primitives that allow for multiple simultaneous programming models.

Vision: Adaptable Programming Model Support
• Support for multiple parallel programming models
• Provide hardware primitives rather than solutions
• Provide programming model support through run-time adaptable Network Interfaces (NI)
  ‒ Handle common cases in hardware
  ‒ Handle corner cases in software
  ‒ This has a cost: Performance impact, NI complexity
• Benefits: Increased flexibility
  ‒ Application specific programming model optimizations
  ‒ Also, better control of the memory system

Architecture Overview
Processor tiles interconnected by a Network on Chip (NoC) as shown in Fig. 1. Key features:
• Processor tile (P)
  ‒ Processor core, Caches, NI, Scratchpad memory
• NoC interconnect
  ‒ Packet switched, mesh topology, source routed
• Network interface
  ‒ Programmable, specialized datapath
  ‒ Tightly integrated with cache controller, processor core
  ‒ co-processor interface and NoC
  ‒ Cache hits in hardware, misses in NI software

Exploiting Adaptable Programming Model Support
• Many tiles - applications own subset of tiles exclusively
  ‒ Simplifies the operating system and task scheduling
• Support for per-application programming models
  ‒ No need to involve tiles that are not a part the application
  ‒ Simultaneous support for shared memory and message passing is possible
  ‒ Enables the possibility of run-time tuning

Evaluation and Results
• Case study: Cache coherent shared memory
  ‒ Directory-based MOESI protocol based on the GEMS-implementation [4]
• Compares two implementations of the protocol in Fig. 2
  ‒ Ideal single cycle hardware
  ‒ Our approach: Adaptable programming model support
• Conservative analytical model of the execution time overhead
• Ongoing simulation work indicates that the model is too conservative. Overhead is likely to be half of the analysis estimates!

Contact
Morten S. Rasmussen, Sven Karlsson, and Jens Sparsø
{msr,ska,jsp}@imm.dtu.dk
Technical University of Denmark
Denmark

References